Non-interactive ADAS and fundamental data

Martin O'Mullane
Department of Physics
University of Strathclyde

Fundamental data production and ADAS

One of the philosophies of ADAS is to provide baseline quality data for atomic processes of any ion stage of an arbitrary element.

These are selectively updated with higher quality data.

Fundamental data production codes available within ADAS:

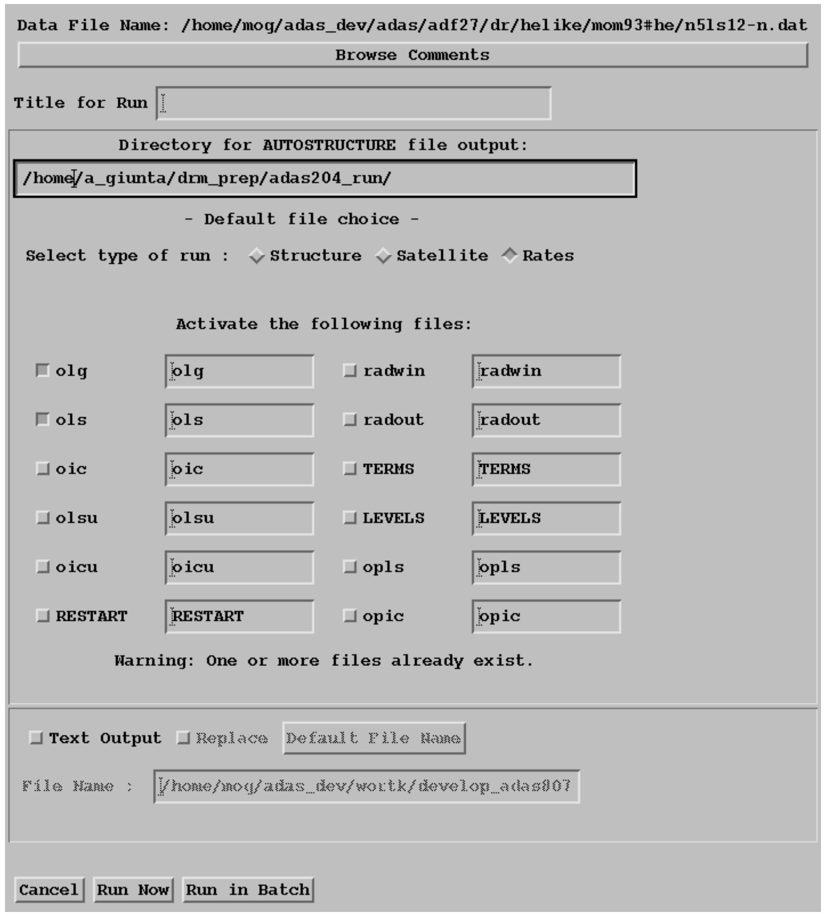
- Cowan code to produce *adf04* data.
- AUTOSTUCTURE for dielectronic recombination. *
- CADW ionisation rate generation.
- R-matrix for electron impact excitation data. *
- IDL and FORTRAN routines for ECIP, Lodge ion impact and others.

- For charge exchange data extraction from a universal formula.

* Difficult to class these as a baseline quality code.

adas701 — AUTOSTRUCTURE
(for dielectronic recombination)

Developed by Nigel Badnell and based on work by W Eissner this (originally a structure code) is tuned to produce state selective dielectronic data.



Data File Name: /home/mog/adas_dev/adas/adf27/dr/helike/mom93#he/n5ls12-n.dat

Browse Comments

Title for Run

Directory for AUTOSTRUCTURE file output:

/home/a_giunta/drm_prep/adas204_run/

- Default file choice -

Select type of run : Structure Satellite [✓ Rates]

Activate the following files:

✓ olg	olg	radwin	radwin
✓ ols	ols	radout	radout
oic	oic	TERMS	TERMS
olsu	olsu	LEVELS	LEVELS
oicu	oicu	opls	opls
RESTART	RESTART	opic	opic

Warning: One or more files already exist.

Text Output Replace Default File Name

File Name : /home/mog/adas_dev/wortk/develop_adas807

Cancel Run Now Run in Batch

Data from adas701

- AUTOSTRUCTURE produces the data for the DR Project.
- 13 isoelectronic sequences, elements up to Zn, IC and LS resolutions.
- 1.2Gb data in *adf09* collection.
- Sets very high bar for ‘baseline’ data.
- See N R Badnell *et al*, ‘Dielectronic recombination data for dynamic finite-density plasmas’, Astron & Astrophys., 406, 1151–1165 (2003).
- Extended for photo-excitation and ionisation.

Input to adas701

A little daunting so reading the manual is essential.

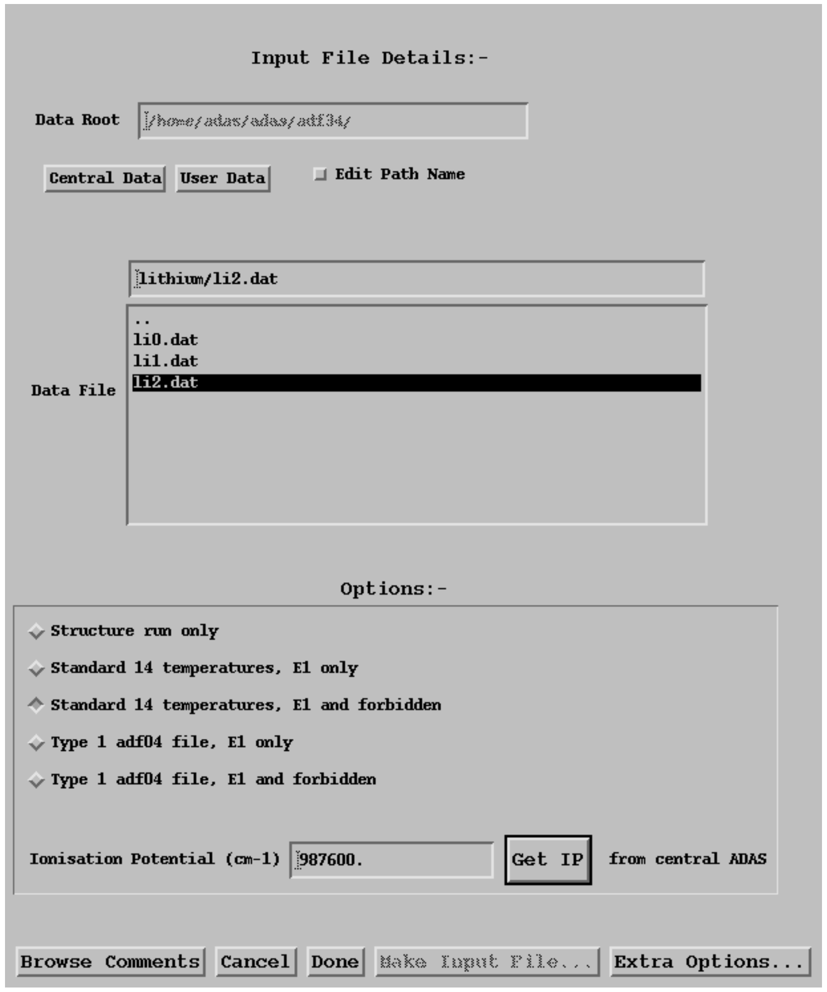
From `/home/adas/adas/adf27/dr/olike/oiz00#o/cu12ic23-n.dat`:

```
S.S.
123456789 22533514517 22533515517 22533516517
          12543514517 12543515517 12543516517
          22543518 22543519 2254351A 2254351B 2254351C
          12553518 12553519 1255351A 1255351B 1255351C
          22533514518 22533514519 2253351451A 2253351451B 2253351451C
          22533515518 22533515519 2253351551A 2253351551B 2253351551C
          22533516518 22533516519 2253351651A 2253351651B 2253351651C
          12543514518 12543514519 1254351451A 1254351451B 1254351451C
          12543515518 12543515519 1254351551A 1254351551B 1254351551C
          12543516518 12543516519 1254351651A 1254351651B 1254351651C
          22543514 22543515 22543516 22543517
          12553514 12553515 12553516 12553517
          22553 12563
               7107207217307317328029009019029003904

 &SALGEB  RUN='DR' RAD='YES' CUP='IC' KORB1=1 KORB2=1 MSTART=4 &END
 &DRR     NMIN=4 NMAX=15 JND=14 LMIN=0 LMAX=6 LCON=5 &END
    16   20   25   35   45   55   70  100  150  200  300  450  700  999
 &SMINIM  NZION=29 PRINT='UNFORM'   &END
 &SRADWIN  KEY=-9  &END
 &SRADCON MENG=-15 &END
    0.0000 160.0000
```

adas801 — Cowan code for adf04 production

This was the first fundamental data generation code added to the structure and forms the basis of series 8.



Input File Details:-

Data Root /home/adas/adas/adf34/

Central Data | User Data | Edit Path Name

Data File lithium/li2.dat

```
..
li0.dat
li1.dat
li2.dat
```

Options:-

- Structure run only
- Standard 14 temperatures, E1 only
- [✓ Standard 14 temperatures, E1 and forbidden]
- Type 1 adf04 file, E1 only
- Type 1 adf04 file, E1 and forbidden

Ionisation Potential (cm-1) 987600. | Get IP | from central ADAS

Browse Comments | Cancel | Done | Make Input File... | Extra Options...

adas8#1 — Cowan code for adf04 production

The # indicates an offline code.

- Designed to be independent of the interactive system (and hence IDL).
- Self-contained to be portable to large computer systems.
- Workhorse code for the heavy species project.

```
/home/adas/offline_adas/adas8#1/scripts/run_adas8#1 c2.in c2.inst c2.pp
```

The input file can be crafted by hand or IDL based tools from the heavy species project can be used.

As an example consider Sn^{13+}

- What is its ground state configuration?
- What configurations contribute to spectral emission?
- And to radiated power?
- How do we choose which ones to include?

The *adf00* set archives ionisation potential and ground configurations:

```
tin                  -50
 0 7.343d+00 1s2  2s2  2p6  3s2  3p6  3d10 4s2  4p6  4d10 4f0  5s2  5p2
 1 1.463d+01 1s2  2s2  2p6  3s2  3p6  3d10 4s2  4p6  4d10 4f0  5s2  5p1
                            ..
                            ..
12 2.744d+02 1s2  2s2  2p6  3s2  3p6  3d10 4s2  4p6  4d2
13 2.995d+02 1s2  2s2  2p6  3s2  3p6  3d10 4s2  4p6  4d1
14 3.959d+02 1s2  2s2  2p6  3s2  3p6  3d10 4s2  4p6
                                ..
```

What configurations should be considered?

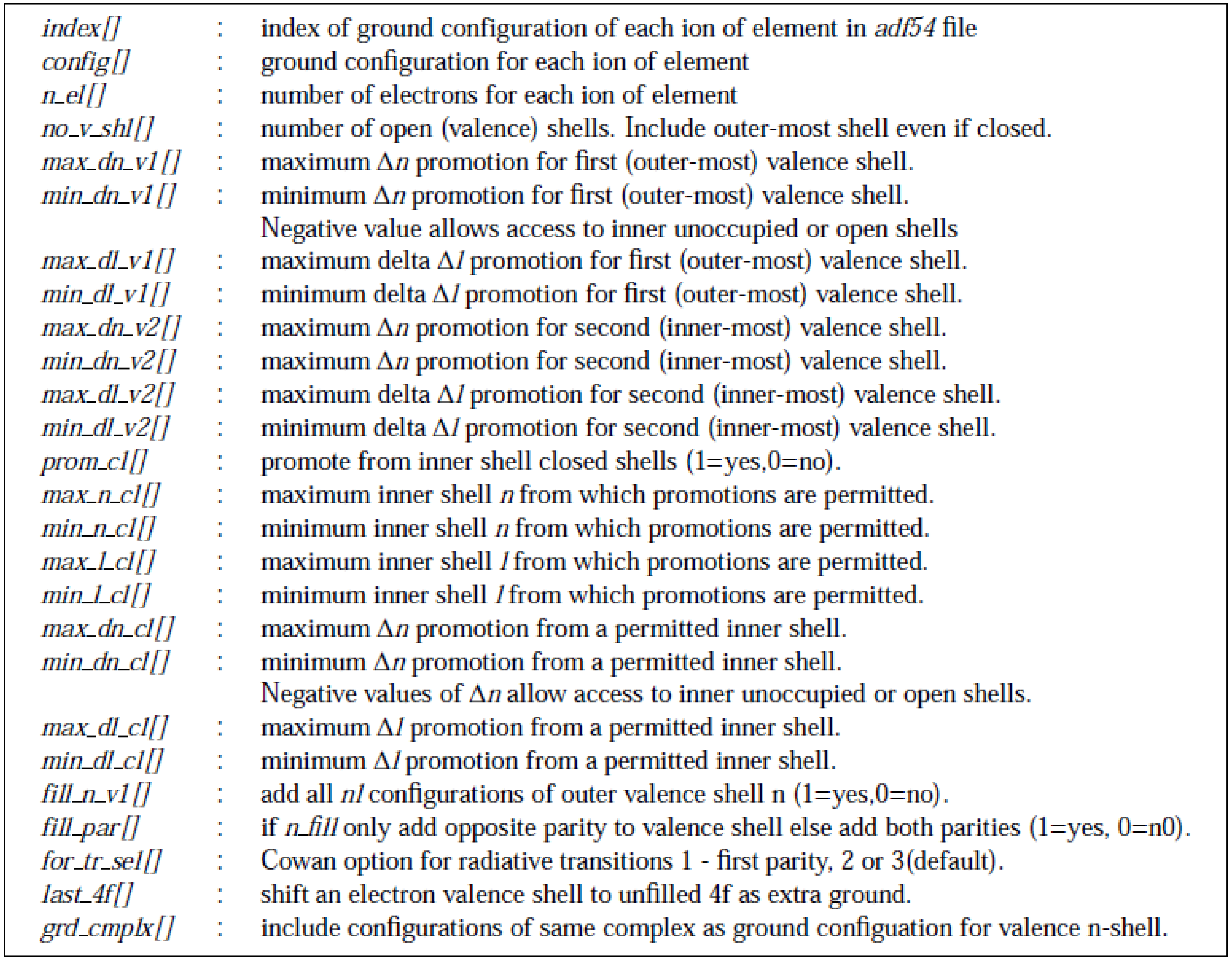
With a ground state of $3\mathrm{d}^{10}4\mathrm{s}^{2}4\mathrm{p}^{6}4\mathrm{d}^{1}$ we can

- promote the valence 4d electron to any higher nl shell
- allow 4s or 4p electrons to be excited
- or any other electron — from 2p perhaps?
- however where do we stop in Δn or Δl?
- and how many configurations should we consider?

There are 180 distinct ground configurations (for elements up to Radon)

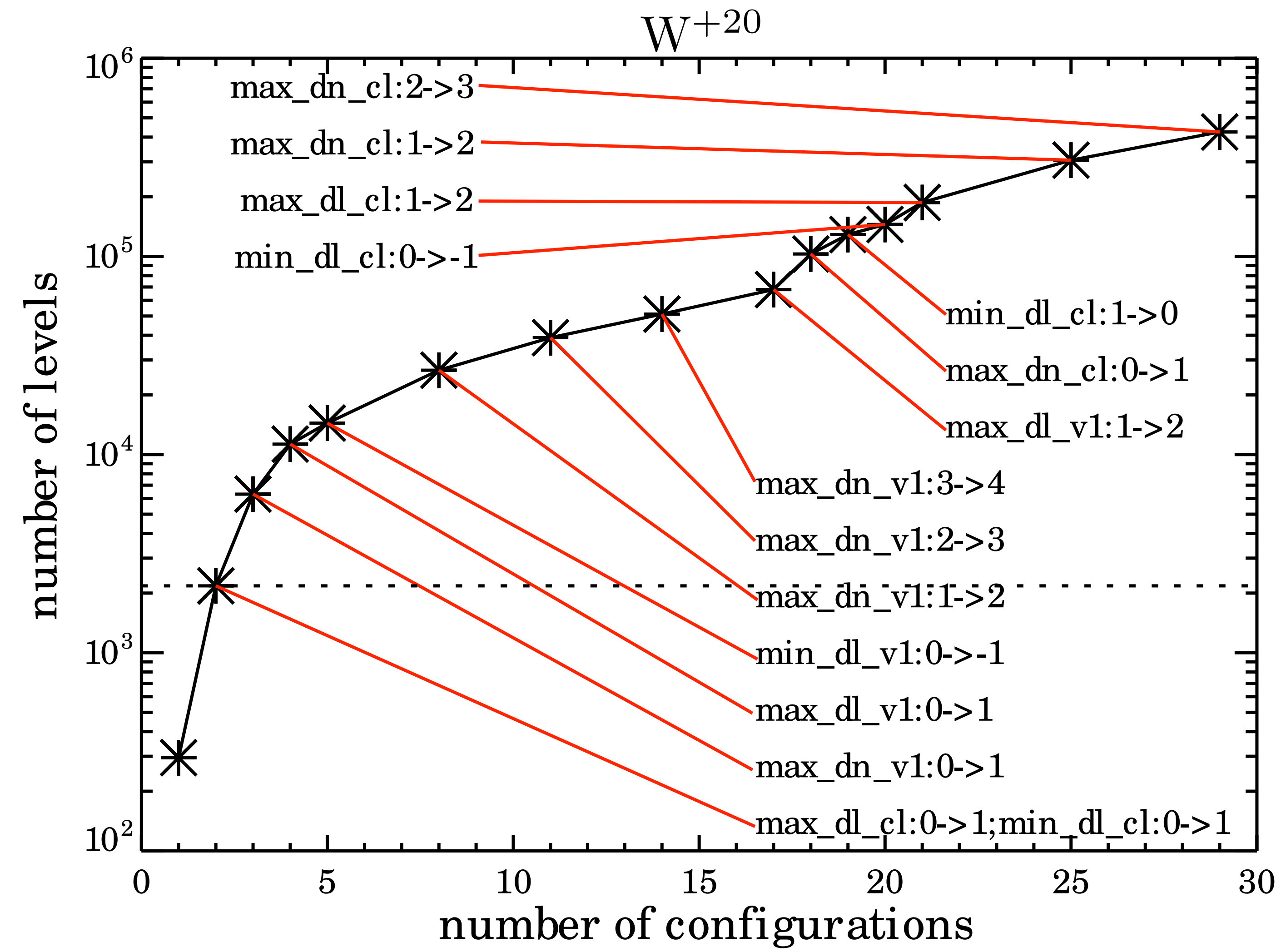
A rule based method is desirable (essential!)

ADAS rules for choosing where to promote electrons



index[]	:	index of ground configuration of each ion of element in *adf54* file
config[]	:	ground configuration for each ion of element
n_el[]	:	number of electrons for each ion of element
no_v_shl[]	:	number of open (valence) shells. Include outer-most shell even if closed.
max_dn_v1[]	:	maximum Δn promotion for first (outer-most) valence shell.
min_dn_v1[]	:	minimum Δn promotion for first (outer-most) valence shell.
		Negative value allows access to inner unoccupied or open shells
max_dl_v1[]	:	maximum delta Δl promotion for first (outer-most) valence shell.
min_dl_v1[]	:	minimum delta Δl promotion for first (outer-most) valence shell.
max_dn_v2[]	:	maximum Δn promotion for second (inner-most) valence shell.
min_dn_v2[]	:	maximum Δn promotion for second (inner-most) valence shell.
max_dl_v2[]	:	maximum delta Δl promotion for second (inner-most) valence shell.
min_dl_v2[]	:	minimum delta Δl promotion for second (inner-most) valence shell.
prom_cl[]	:	promote from inner shell closed shells (1=yes,0=no).
max_n_cl[]	:	maximum inner shell n from which promotions are permitted.
min_n_cl[]	:	minimum inner shell n from which promotions are permitted.
max_l_cl[]	:	maximum inner shell l from which promotions are permitted.
min_l_cl[]	:	minimum inner shell l from which promotions are permitted.
max_dn_cl[]	:	maximum Δn promotion from a permitted inner shell.
min_dn_cl[]	:	minimum Δn promotion from a permitted inner shell.
		Negative values of Δn allow access to inner unoccupied or open shells.
max_dl_cl[]	:	maximum Δl promotion from a permitted inner shell.
min_dl_cl[]	:	minimum Δl promotion from a permitted inner shell.
fill_n_v1[]	:	add all nl configurations of outer valence shell n (1=yes,0=no).
fill_par[]	:	if *n_fill* only add opposite parity to valence shell else add both parities (1=yes, 0=n0).
for_tr_sel[]	:	Cowan option for radiative transitions 1 - first parity, 2 or 3(default).
last_4f[]	:	shift an electron valence shell to unfilled 4f as extra ground.
grd_cmplx[]	:	include configurations of same complex as ground configuation for valence n-shell.

adf54 : rules for automatic data generation



Care needed!! resolved calculations (ic or LS) can overwhelm computers.

Work through Sn^{13+}

- Within ADAS the generation of heavy species data is almost exclusively a non-GUI activity.
- The outputs are standard *adf11*, *adf15* and *adf40* datasets which can be used and examined with the GUI interactive system.

At the IDL command line:

```
; Let's choose Sn13+

z_nuc = 50
z_ion = 13
tag   = xxesym(z_nuc, /lower) + string(z_ion, format='(i2.2)')

; Use promotion rules from W work

a54file = '/home/adas/adas/adf54/promotion_rules_w_adf54.dat'
```

```
adas8xx_promotion_rules, z0_nuc = z_nuc, z_ion = z_ion, ionpot = ip,  $
                         prom_rules=rules, a54file = file
help, rules, /st

** Structure <9b54e9c>, 25 tags, length=60, data length=60, refs=1:
   CONFIG        STRING '  1s2  2s2  2p6  3s2  3p6  3d10 4s2  4p6  4d1'
   INDEX         INT            129
   NO_V_SHL      INT              1
   MAX_DN_V1     INT              3
   MIN_DN_V1     INT              0
   MAX_DL_V1     INT              2
   MIN_DL_V1     INT             -2
   MAX_DN_V2     INT              0
   MIN_DN_V2     INT              0
   MAX_DL_V2     INT              0
   MIN_DL_V2     INT              0
   PROM_CL       INT              1
   MAX_N_CL      INT              4
   MIN_N_CL      INT              4
```

```
MAX_L_CL     INT            1
MIN_L_CL     INT            0
MAX_DN_CL    INT            1
MIN_DN_CL    INT            0
MAX_DL_CL    INT            2
MIN_DL_CL    INT            0
FILL_N_V1    INT            1
FILL_PAR     INT            0
FOR_TR_SEL   INT            3
LAST_4F      INT            0
GRD_CMPLX    INT            0
```

```
adas8xx_promotions, z0_nuc = z_nuc, z_ion = z_ion, ionpot = ip,  $
                    prom_rules        = rules,                    $
                    promotion_results = results

help, results, /st

** Structure <9b530dc>, 11 tags, length=2496, data length=2496, refs=1:
   GRD_CFG         STRING    '4d1  '
   GRD_OCC         INT       Array[36]
   EX_CFG          STRING    Array[25]
   GRD_PAR         INT              0
   EX_PAR          INT       Array[25]
   GRD_ZC_COW      LONG               -14
   EX_ZC_COW       LONG      Array[25]
   OC_STORE        INT       Array[36, 26]
   NO_CONFIGS      LONG      Array[7]
   NO_TERMS        LONG      Array[7]
   NO_LEVELS       LONG      Array[7]
```

```
print, results.grd_occ

        2       2       6       2       6      10       2       6       1       0
        0       0       0       0       0       0       0       0       0       0
        0       0       0       0       0       0       0       0       0       0
        0       0       0       0       0       0

print, results.oc_store[*,1]
        2       2       6       2       6      10       2       6       0       1
        0       0       0       0       0       0       0       0       0       0
        0       0       0       0       0       0       0       0       0       0
        0       0       0       0       0       0

print, results.oc_store[*,2]
        2       2       6       2       6      10       2       6       0       0
        1       0       0       0       0       0       0       0       0       0
        0       0       0       0       0       0       0       0       0       0
        0       0       0       0       0       0
```

```
; Write CA driver files for restricted plasma parameters

files = { adf34_file    : 'adf34_ic_' + tag + '.dat', $
          adf42_ic_file : 'adf42_ic_' + tag + '.dat', $
          adf04_ic_file : 'adf04_ic_' + tag + '.dat', $
          adf40_ic_file : 'adf40_ic_' + tag + '.dat', $
          adf15_ic_file : 'adf15_ic_' + tag + '.dat', $
          adf11_ic_file : 'adf11_ic_' + tag + '.dat'}

plasma = {theta         : [  1.0e3, 2.0e3, 5.0e3, 1.0e4, 1.5e4, $
                             2.0e4, 5.0e4, 1.0e5], $
          indx_theta    : indgen(8),
          rho           : [  1.0e8, 1.0e10, 1.0e12, 1.0e14],
          indx_rho      : indgen(4),
          npix          : [  128,   256],
          wvlmin        : [100.0,   1.0],
          wvlmax        : [150.0, 500.0],
          indx_wvl      : indgen(2),
          theta_noscale : 0,
          rho_scale     : 0
```

```
adas8xx_create_drivers, z0_nuc=z_nuc, z_ion=z_ion, ionpot=ip, $
                        promotion_results=results,            $
                        plasma=plasma, files=files
```

The driver file for ADAS801 (Cowan code):

```
2  -5    2   10  1.0    5.d-09    5.d-11-2  0130    1.0 0.65  0.0  0.5
   50  -14   Sn ground z1=13 0   4d1
   50  -14   Sn cfg 01       0   5s1
   50  -14   Sn cfg 02       0   5d1
   50  -14   Sn cfg 03       0   5g1
   50  -14   Sn cfg 04       0   6s1
   50  -14   Sn cfg 05       0   6d1
   50  -14   Sn cfg 06       0   6g1
   50  -14   Sn cfg 07       0   7s1
   50  -14   Sn cfg 08       0   7d1
   50  -14   Sn cfg 09       0   7g1
   50  -32   Sn cfg 10       0   3d10 4s1  4p6  4d2
   50  -32   Sn cfg 11       0   3d10 4s1  4p6  4d1  5s1
```

```
50  -32   Sn cfg 12        0   3d10 4s1  4p6  4d1  5d1
50  -32   Sn cfg 13        0   3d10 4s2  4p5  4d1  4f1
50  -32   Sn cfg 14        0   3d10 4s2  4p5  4d1  5p1
50  -32   Sn cfg 15        0   3d10 4s2  4p5  4d1  5f1
50  -14   Sn cfg 16        1   4f1
50  -14   Sn cfg 17        1   5p1
50  -14   Sn cfg 18        1   5f1
50  -14   Sn cfg 19        1   6p1
50  -14   Sn cfg 20        1   6f1
50  -14   Sn cfg 21        1   7p1
50  -14   Sn cfg 22        1   7f1
50  -32   Sn cfg 23        1   3d10 4s1  4p6  4d1  5p1
50  -32   Sn cfg 24        1   3d10 4s2  4p5  4d2
50  -32   Sn cfg 25        1   3d10 4s2  4p5  4d1  5d1
-1
```

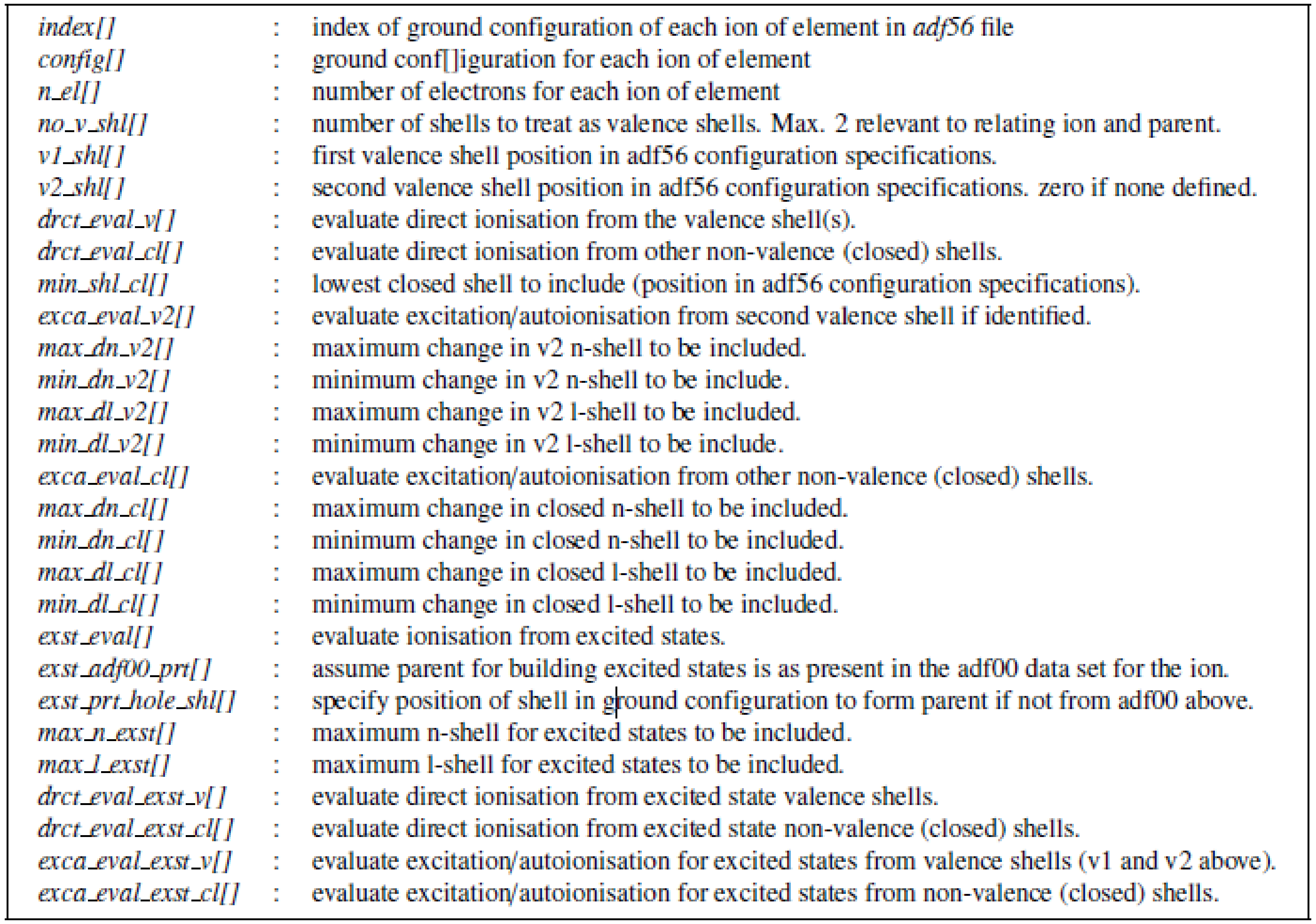
Limitations of adas801/adas8#1

- The Cowan code is well integrated into ADAS workflows.
- adas8#1 has been extended to incorporate as optional inputs the U Mons improved structure work.
- However.... spin changing transitions are absent.
- No resonance effects.

- AUTOSTRUCTURE now has a distorted wave module and can generate *adf04* datasets.
- Auto-generation of input files will be added to the heavy species rules-based routines.
- Under active development with data already produced (HPS).

adas8#2 — CADW Ionisation

Very similar specification probelm as excitation — driven by *adf56* set of rules



index[]	:	index of ground configuration of each ion of element in *adf56* file
config[]	:	ground conf[]iguration for each ion of element
n_el[]	:	number of electrons for each ion of element
no_v_shl[]	:	number of shells to treat as valence shells. Max. 2 relevant to relating ion and parent.
v1_shl[ ]	:	first valence shell position in adf56 configuration specifications.
v2_shl[ ]	:	second valence shell position in adf56 configuration specifications. zero if none defined.
drct_eval_v[ ]	:	evaluate direct ionisation from the valence shell(s).
drct_eval_cl[ ]	:	evaluate direct ionisation from other non-valence (closed) shells.
min_shl_cl[]	:	lowest closed shell to include (position in adf56 configuration specifications).
exca_eval_v2[]	:	evaluate excitation/autoionisation from second valence shell if identified.
max_dn_v2[]	:	maximum change in v2 n-shell to be included.
min_dn_v2[ ]	:	minimum change in v2 n-shell to be include.
max_dl_v2[]	:	maximum change in v2 l-shell to be included.
min_dl_v2[ ]	:	minimum change in v2 l-shell to be include.
exca_eval_cl[ ]	:	evaluate excitation/autoionisation from other non-valence (closed) shells.
max_dn_cl[]	:	maximum change in closed n-shell to be included.
min_dn_cl[ ]	:	minimum change in closed n-shell to be included.
max_dl_cl[ ]	:	maximum change in closed l-shell to be included.
min_dl_cl[ ]	:	minimum change in closed l-shell to be included.
exst_eval[]	:	evaluate ionisation from excited states.
exst_adf00_prt[ ]	:	assume parent for building excited states is as present in the adf00 data set for the ion.
exst_prt_hole_shl[]	:	specify position of shell in ground configuration to form parent if not from adf00 above.
max_n_exst[]	:	maximum n-shell for excited states to be included.
max_l_exst[]	:	maximum l-shell for excited states to be included.
drct_eval_exst_v[ ]	:	evaluate direct ionisation from excited state valence shells.
drct_eval_exst_cl[ ]	:	evaluate direct ionisation from excited state non-valence (closed) shells.
exca_eval_exst_v[]	:	evaluate excitation/autoionisation for excited states from valence shells (v1 and v2 above).
exca_eval_exst_cl[]	:	evaluate excitation/autoionisation for excited states from non-valence (closed) shells.

adf32 is the driver file for CADW ionisation code from the Auburn group.

At the IDL command line

```
; Add offline-ADAS IDL library to the path

!path = expand_path('/home/adas/offline_adas/adas8#2/idl') + ':' + !path

; Promotion rules - compiled by Adam Foster (arf)

a56file = '/home/adas/adas/adf56/large_arf09.dat'

; Sn13+ !!

adas8xx_ionis_promotion_rules, z_nuc    = 50,                    $
                               z_ion    = 13,                    $
                               a56file  = a56file,               $
                               adf32    = 'adf32_ca_sn13.dat',   $
                               comments = ['C--------------------', $
                                           'C  I made this!',     $
                                           'C--------------------'  ]
```

```
elem   = Sn
stage  = 13
ip_z   =    3193147.3
ip_z1  =    2415629.2
seq    = rb
--------------------------------------------------------------------------------
Type = Direct /number=3/
#
200-51 1 2  01.   1.   5.0E-08   1.0E-11-2  0130 0 1.00 0.65   71.  0.5       0.70
   50   14  sn+13 ground     4d1                                     4d
   50   15  sn+14 from 4d    3d10 4s2  4p6
   -1
#
200-51 1 2  01.   1.   5.0E-08   1.0E-11-2  0130 0 1.00 0.65   73.  0.5       0.70
   50   14  sn+13 ground     4d1                                     4s
   50   15  sn+14 from 4s    3d10 4s1  4p6  4d1
   -1
#
200-51 1 2  01.   1.   5.0E-08   1.0E-11-2  0130 0 1.00 0.65   72.  0.5       0.70
   50   14  sn+13 ground     4d1                                     4p
   50   15  sn+14 from 4p    3d10 4s2  4p5  4d1
   -1
--------------------------------------------------------------------------------
Type = InDirect /number=2/
#
20 -51 0 2   10  1.0    5.e-08    1.e-11-2   130    1.0 0.65   66.   0.5       0.7
   50   14  sn+13 ground     4d1                                     4s
   50   14  sn+13 via  4d    3d10 4s1  4p6  4d2                      4d
   50   14  sn+13 via  4f    3d10 4s1  4p6  4d1  4f1                 4f
   50   14  sn+13 via  5s    3d10 4s1  4p6  4d1  5s1                 5s
                               -
                               -
   50   14  sn+13 via  7h    3d10 4s1  4p6  4d1  7h1                 7h
   50   14  sn+13 via  7i    3d10 4s1  4p6  4d1  7i1                 7i
   -1
#
20 -51 0 2   10  1.0    5.e-08    1.e-11-2   130    1.0 0.65   66.   0.5       0.7
   50   14  sn+13 ground     4d1                                     4p
   50   14  sn+13 via  4d    3d10 4s2  4p5  4d2                      4d
   50   14  sn+13 via  4f    3d10 4s2  4p5  4d1  4f1                 4f
                               -
                               -
   50   14  sn+13 via  7h    3d10 4s2  4p5  4d1  7h1                 7h
   50   14  sn+13 via  7i    3d10 4s2  4p5  4d1  7i1                 7i
   -1
--------------------------------------------------------------------------------
C-------------------
C  I made this!
C-------------------
```

At the unix command line

```
 /home/adas/offline_adas/adas8#2/adas8#2.pl \
      adf32_ca_sn13.dat adf23_ca_sn13.dat
```

Return to IDL to inspect the results

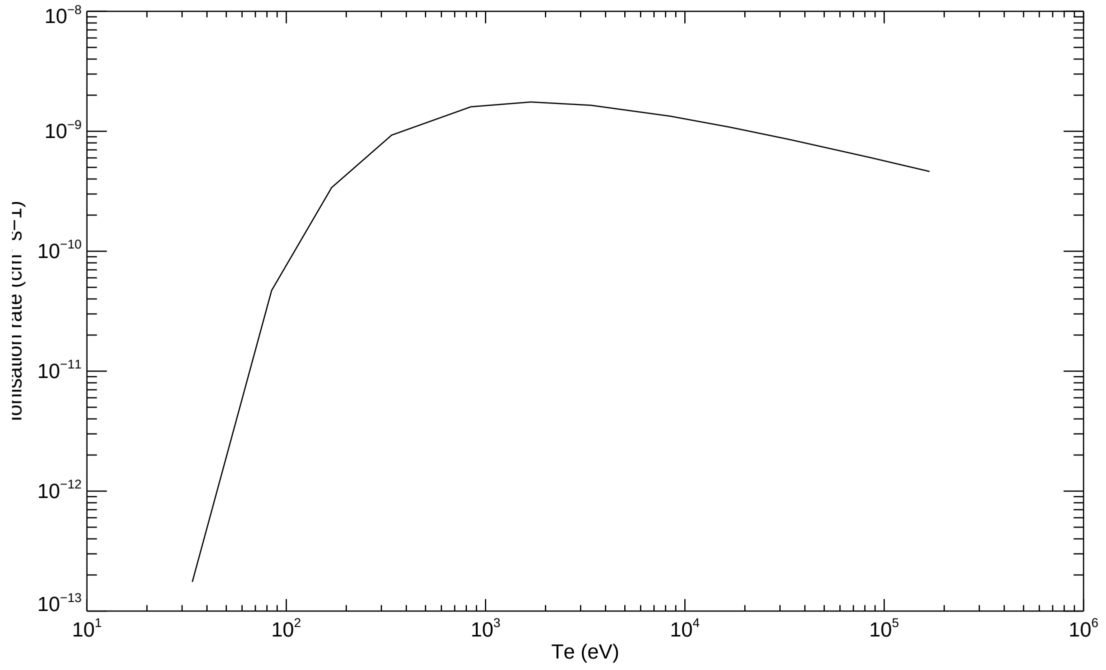
```
read_adf23, file='adf23_ca_sn13.dat', fulldata=all, szd_total=szd

help, szd,/st

** Structure <a3e784c>, 7 tags, length=6576, data length=6576, refs=1:
   TE               DOUBLE    Array[12]
   Q_ION            DOUBLE    Array[1, 3, 12]
   IS_Q_ION         LONG      Array[1, 3, 12]
   Q_EXC            DOUBLE    Array[1, 41, 12]
   IS_Q_EXC         LONG      Array[1, 41, 12]
   QTOT             DOUBLE    Array[1, 1, 12]
   IS_QTOT          LONG      Array[1, 1, 12]
```

```
te  = reform(szd.te) / 11605.0
szd = reform(szd.qtot*10.0^szd.is_qtot) > 1.0e-36

plot_oo, te, szd,           $
         xtitle='Te (eV)', $
         ytitle = 'Ionisation rate (cm!u3!n s-1!n)
```



Uplift of baseline

As codes mature they can transition from the domain of the specialist to routine, unattended mass data generation. Standard R-matrix is now at this stage — adas8#3.

```
/home/adas/offline_adas/adas8#3/scripts/adas8#3.pl input.dat Z
```

```
GENERAL                            SCALING PARAMETERS
2Jmax_ex = 23                      1s = 1.0
2Jmax_nx = 91                      2s = 1.0
maxc = 51                          2p = 1.0
mesh_fine = 0.0025                 3s = 1.0
mesh_coarse = 0.01                 3p = 1.0
maxe/ionpot = 3                    3d = 1.0
rdamp = 1
adamp = 0

CONFIGURATION LIST
1s2
1s1 2s1
1s1 2p1
1s1 3s1
1s1 3p1
1s1 3d1
```